```
entry:
 call void @llvm.dbg.value(metadata ptr %a, metadata !21, metadata
... !DIExpression()), !dbg !27
 call void @llvm.dbg.value(metadata ptr %b, metadata !22, metadata
... !DIExpression()), !dbg !27
 call void @llvm.dbg.value(metadata ptr %c, metadata !23, metadata
... !DIExpression()), !dbg !27
 call void @llvm.dbg.value(metadata i32 %n, metadata !24, metadata
... !DIExpression()), !dbg !27
 tail call void asm sideeffect "dmb sy\0A\09orr x3,x3,x3\0A", "~{memory}"()
... #8, !dbg !28, !srcloc !29
 call void @llvm.dbg.value(metadata i32 0, metadata !25, metadata
... !DIExpression()), !dbg !30
 %cmp11 = icmp sgt i32 %n, 0, !dbg !31
 br i1 %cmp11, label %for.body.preheader, label %for.cond.cleanup, !dbg !33
```

```
for.body.preheader:
 %wide.trip.count = zext i32 %n to i64, !dbg !31
 %0 = call i64 @llvm.vscale.i64(), !dbg !33
 %1 = shl i64 %0, 2, !dbg !33
 %2 = icmp uge i64 %wide.trip.count, %1, !dbg !33
 br i1 %2, label %Pre.Vectorization, label
... %Preheader.for.remaining.iterations, !dbg !33
```

```
Pre.Vectorization:
 %5 = call <vscale x 4 x i64> @llvm.experimental.stepvector.nxv4i64()
 %6 = call i64 @llvm.vscale.i64()
 %step.value = shl i64 %6, 2
 %7 = urem i64 %wide.trip.count, %step.value
 %total.iterations.to.be.vectorized = sub i64 %wide.trip.count, %7
 %8 = insertelement <vscale x 4 x i64> poison, i64 %step.value, i64 0
 %stepVector.update.values = shufflevector <vscale x 4 x i64> %8, <vscale x 4
... x i64> poison, <vscale x 4 x i32> zeroinitializer
 br label %vectorizing.block
```

```
vectorizing.block:
 %9 = phi i64 [ 0, %Pre.Vectorization ], [ %20, %vectorizing.block ]
 %10 = phi <vscale x 4 x i64> [ %5, %Pre.Vectorization ], [ %21,
... %vectorizing.block ]
 %11 = and <vscale x 4 x i64> %10, shufflevector (<vscale x 4 x i64>
... insertelement (<vscale x 4 x i64> poison, i64 1, i64 0), <vscale x 4 x i64>
... poison, <vscale x 4 x i32> zeroinitializer)
 %12 = icmp eq <vscale x 4 x i64> %11, zeroinitializer
 %13 = bitcast <vscale x 4 x i1> %12 to <vscale x 4 x i1>
 %14 = getelementptr inbounds i32, ptr %a, i64 %9, !dbg !41
 %15 = getelementptr inbounds i32, ptr %b, i64 %9, !dbg !47
 %16 = getelementptr inbounds i32, ptr %c, i64 %9, !dbg !49
 %17 = call <vscale x 4 x i32> @llvm.aarch64.sve.ld1.nxv4i32(<vscale x 4 x
... i1> %13, ptr %14)
 %18 = call <vscale x 4 x i32> @llvm.aarch64.sve.ld1.nxv4i32(<vscale x 4 x
... i1> %13, ptr %15)
 %19 = call <vscale x 4 x i32> @llvm.aarch64.sve.mul.nxv4i32(<vscale x 4 x
... i1> %13, <vscale x 4 x i32> %18, <vscale x 4 x i32> %17)
 call void @llvm.aarch64.sve.st1.nxv4i32(<vscale x 4 x i32> %19, <vscale x 4
... x i1> %13, ptr %16)
 %20 = add i64 %step.value, %9
 %21 = add <vscale x 4 x i64> %10, %stepVector.update.values
 %terminate.condition = icmp uge i64 %20, %total.iterations.to.be.vectorized
 br i1 %terminate.condition, label %middle.block, label %vectorizing.block
```

```
middle.block:
 %condition = icmp eq i64 %7, 0
 br i1 %condition, label %for.cond.cleanup.loopexit, label
... %Preheader.for.remaining.iterations
```

```
Preheader.for.remaining.iterations:
 %22 = phi i64 [ 0, %for.body.preheader ], [ %20, %middle.block ]
 br label %for.body
```

```
for.body:
 %indvars.iv = phi i64 [ %indvars.iv.next, %for.inc ], [ %22,
... %Preheader.for.remaining.iterations ]
 call void @llvm.dbg.value(metadata i64 %indvars.iv, metadata !25, metadata
... !DIExpression()), !dbg !30
 %rem15 = and i64 %indvars.iv, 1, !dbg !37
 %cmp1.not = icmp eq i64 %rem15, 0, !dbg !37
 br i1 %cmp1.not, label %for.inc, label %if.then, !dbg !40
```

```
if.then:
 %arrayidx = getelementptr inbounds i32, ptr %a, i64 %indvars.iv, !dbg !41
 %3 = load i32, ptr %arrayidx, align 4, !dbg !41, !tbaa !43
 %arrayidx3 = getelementptr inbounds i32, ptr %b, i64 %indvars.iv, !dbg !47
 %4 = load i32, ptr %arrayidx3, align 4, !dbg !47, !tbaa !43
 %mul = mul nsw i32 %4, %3, !dbg !48
 %arrayidx5 = getelementptr inbounds i32, ptr %c, i64 %indvars.iv, !dbg !49
 store i32 %mul, ptr %arrayidx5, align 4, !dbg !50, !tbaa !43
 br label %for.inc, !dbg !51
```

```
for.inc:
 %indvars.iv.next = add nuw nsw i64 %indvars.iv, 1, !dbg !52
 call void @llvm.dbg.value(metadata i64 %indvars.iv.next, metadata !25,
... metadata !DIExpression()), !dbg !30
 %exitcond.not = icmp eq i64 %indvars.iv.next, %wide.trip.count, !dbg !31
 br i1 %exitcond.not, label %for.cond.cleanup.loopexit, label %for.body, !dbg
... !33, !llvm.loop !53
```

```
for.cond.cleanup.loopexit:
 br label %for.cond.cleanup, !dbg !34
```

```
for.cond.cleanup:
 tail call void asm sideeffect "dmb sy\0A\09orr x4,x4,x4\0A", "~{memory}"()
... #8, !dbg !34, !srcloc !35
 ret void, !dbg !36
```

CFG for 'foo' function